Accelerated Volume Graphics

Arie E. Kaufman
Computer Science Department
Stony Brook University
Stony Brook, NY 11794-4400, USA
ari@cs.sunysb.edu

Abstract

This paper introduces the field of volume visualization, volumetric data representations, and volume rendering algorithms. It further discusses volume graphics and its underlying voxelization algorithms. Special-purpose volume rendering architectures have been researched for over two decades. Recently, commercial real-time volume rendering boards have been introduced, most notably the VolumePro board which is based on the Cube-4 architecture developed at Stony Brook University.

1. Volume visualization

Volume visualization is a method of extracting information from volumetric datasets through interactive graphics and imaging, and is concerned with the representation, manipulation, and rendering of these datasets [13]. Volume data are 3D entities that may have information inside them, might not consist of surfaces and edges, or might be too voluminous to be represented geometrically. Volume visualization encompasses an array of techniques for peering inside the dataset and for interactively extracting meaningful information from it using transformations, cuts, segmentation, translucency, measurements, navigation, and the like. The primary sources of volume data are three: sampled data of real objects or phenomena, computed data produced by a computer simulation, or modeled data generated from a geometric model. Examples of applications generating sampled data are medical imaging (e.g., CT, MRI), biology (e.g., confocal microscopy), geoscience (e.g., seismic measurements), industry (e.g., non-destructive inspection), and chemistry (e.g., electron density maps) [13]. Some examples of applications generating computed datasets by typically running a simulation on a supercomputer are meteorology (e.g., storm prediction), computational fluid dynamics (e.g., water flow), and material science (e.g., new materials). Recently, many traditional computer graphics applications, such as computer-aided design and flight simulation [3, 28, 31], have been exploiting the advantages of volumetric techniques for modeling, manipulation, and visualization; an approach called *volume graphics* [14].

2. Volumetric data

Volumetric data is typically a set of samples representing the value of some property of the data at a 3D locations. If the value is simply a 0 or a 1, with 0 indicating background and 1 indicating the object, the data is referred to as binary data. The data may instead be multivalued, with the value representing some measurable property of the data, such as density, color, heat, or pressure. The value may even be a vector, representing, for example, velocity at each location. In general, the samples may be taken at random locations in space, but in many cases the dataset is isotropic containing samples taken at regularly spaced intervals along three orthogonal axes. Since dataset is defined on a *regular grid*, a 3D array (called *volume buffer, cubic frame buffer, 3D raster*) is typically used to store the values. the dataset is therefore referred to as the array of values, which is defined only at grid locations. A function may be defined to describe the value at any continuous location by approximating at a location using some interpolation function of the data, such as zero-order (nearest-neighbor), piecewise function known as first-order (trilinear), or higher-order interpolation. The region of constant value that surrounds each sample in zero-order interpolation is known as a volume cell (in short, *voxel*), with each voxel being a rectangular cuboid having six faces, twelve edges, and eight corners. The terms voxel, grid location, and sample points are often used interchangeably.

In addition to regular grids, rectilinear, curvilinear, and unstructured grids are employed. In a *rectilinear*

Proceedings of the Geometric Modeling and Processing — Theory and Applications
0-7695-1674-2/02 $17.00 © 2002 **IEEE**

grid the cells are axis-aligned, but grid spacings along the axes might be arbitrary. When such a grid has been non-linearly transformed while preserving the grid topology, the grid becomes *curvilinear*. Usually, the rectilinear grid defining the logical organization is called *computational space,* and the curvilinear grid is called *physical space*. In an *unstructured grid*, there is no explicit or implicit grid topology. Cells may be, for example, tetrahedra, hexahedra, pyramids, or prisms, where tetrahedral grids are specifically popular. Unstructured grids are common for scattered data, finite element/volume analysis, and computational fluid dynamics.

3. Volume rendering

Over the years many techniques have been developed to visualize volumetric data. Most of the early methods (e.g., Marching Cubes [17]), involve approximating a surface contained within the data using geometric primitives, since methods for displaying geometric primitives were already well-established. When such surface rendering is used, a dimension of information is essentially lost. In addition, adequate approximations may require an excessive amount of primitives. Also, amorphous phenomena, such as clouds, fog, and fire cannot be adequately represented using surfaces. In response to these, *volume rendering* techniques have been developed, attempting to capture the entire 3D data in a single 2D image, by projecting a 2D image directly from the 3D volumetric data. Volume rendering convey more information than surface rendering, but at the cost of increased algorithm complexity, and consequently increased rendering times. To improve interactivity in volume rendering, many optimization methods and several special-purpose volume rendering machines have been developed, as described below.

Volume rendering can be divided into three main approaches: object-order, image-order, and domain methods. In *object-order* methods the contribution of each voxel to the screen pixels is calculated, and the combined contribution yields the final image. One such method is splatting [30]), in which an image-plane footprint for each voxel is used to spread the voxel energy onto a neighborhood of pixels. In *image-order* methods, such as *ray casting*, rays are cast from the screen pixels into the volume, and the contributions of voxels along a ray are calculated and used to color the corresponding pixel (e.g., [16]). In *domain* methods the spatial data is transformed into an alternative domain, such as compression (e.g., [32]), wavelet (e.g., [21]), or frequency domain (e.g., [18]), from which a projection is directly generated.

4. Volume graphics

Although volumetric representation and visualization seem more natural for sampled or computed datasets, their advantages have also been attracting traditional computer graphics applications that deal with synthetic scenes represented by geometric models. In this emerging approach, called *volume graphics*, the geometric model is *voxelized (3D scan-converted)* into a set of voxels that "best" approximate the model. Each of these voxels is stored in the volume buffer along with the voxel pre-computed view-independent attributes, such as texture, anti-aliasing, normal vector, etc. The voxelized model can be either binary (e.g., [4, 9-11]) or volume sampled, [27, 29] which is an alias-free filtered voxelization of the model. In many applications involving sampled or computed data, such as medial imaging, the data need to be visualized along with synthetic objects, such as scalpels, prosthetic devices, mirrors, and radiation beams. These geometric objects can be voxelized and intermixed with the sampled object in the volume buffer.

Volume graphics is concerned with the synthesis, manipulation, and rendering of volumetric geometric objects [14]. Unlike volume visualization which focuses primarily on sampled and computed datasets, volume graphics is concerned primarily with modeled geometric scenes and particularly with those that are represented in a volume buffer. As an approach, volume graphics has the potential to greatly advance the field of 3D graphics by offering a comprehensive alternative to traditional surface graphics. Volume graphics has advantages over surface graphics by being viewpoint independent, insensitive to scene and object complexities, and it lends itself to block operations, constructive solid modeling, and hierarchical representations. It is suitable for the representation of sampled or computed datasets and their intermixing with geometric objects, and it supports the visualization of internal structures and amorphous phenomena. The problems associated with the volumetric representation, such as memory size, processing time, aliasing, and lack of geometric information, echo problems encountered when raster

Proceedings of the Geometric Modeling and Processing — Theory and Applications
0-7695-1674-2/02 $17.00 © 2002 **IEEE**

graphics emerged as an alternative technology to vector graphics, and can be alleviated in similar ways.

5. Hardware for volume rendering

The high computational cost of direct volume rendering makes it difficult for sequential implementations and general-purpose computers to deliver the targeted level of performance. This situation is aggravated by the continuing trend towards higher and higher resolution datasets. For example, to render a dataset of 1024^3 16-bit voxels at 30 frames per second requires 2 GBytes of storage, a memory transfer rate of 60 GBytes per second, and approximately 300 billion instructions per second, assuming merely 10 instructions per voxel per projection. To address this challenge, researchers have tried to achieve interactive display rates on supercomputers and massively parallel architectures. However, most algorithms require very little repeated computation on each voxel and data movement which is inefficient on supercomputers actually accounts for a significant portion of the overall performance overhead.

In the same way that the special requirements of traditional computer graphics lead to high-performance graphics engines, volume visualization naturally lends itself to special-purpose volume renderers which separate real-time image generation from general-purpose processing. This allows for stand-alone visualization environments that help scientists to interactively view their data on a single user workstation, either augmented by a volume rendering accelerator or connected to a dedicated visualization server. Furthermore, a volume rendering engine integrated in a graphics workstation is a natural extension of raster-based systems into 3D volume visualization.

Several researchers have proposed special-purpose volume rendering architectures [13, Chapter 6] [6, 8, 19, 22]. Most recent research focuses on accelerators for ray-casting of regular datasets. Ray-casting offers room for algorithmic improvements while still allowing for high image quality. Recent architectures include VOGUE [15], VIZRAD II [20], VIRIM [7] and VIRIM II [5].

The Cube project at Stony Brook University aims towards the realization of high-performance volume rendering systems for large datasets and has pioneered several hardware architectures based on a specially skewed memory organization which enables conflict-free access to any ray parallel to a main axis. Cube-1, [12] Cube-2 [1], and Cube-3 [24]. The most recent architecture, Cube-4 [25] has only simple and local interconnections. Instead of processing individual rays, Cube-4 manipulates a volume slice at a time. Consequently, Cube-4 is easily scalable to a very high resolution of 1024^3 16-bit voxels and true real-time performance implementations of 30 frames per second.

Enhancing the Cube-4 architecture, Mitsubishi Electric has derived EM-Cube (Enhanced Memory Cube-4). [23]. The primary innovation of EM-Cube is the block-skewed memory, where volume memory is organized in subcubes (blocks) in such a way that all voxels of a block are stored linearly in the same DRAM page. EM-Cube has been further developed into a commercial product where a volume rendering chip, called vg500, has been developed by Mitsubishi. It computes 500 million interpolated, Phong-illuminated, composited samples per second. The vg500 is the heart of a VolumePro500 PC card consisting of one vg500 and configurable standard SDRAM memory architectures. The first generation, available in 1999, supports rendering of a rectangular data set up to $256 \times 256 \times 256$ 12-bit voxels, in real-time 30 frames/sec [26]. Cube-4 Light [2] added perspective projection capabilities to the architecture, but with some view dependent filtering artifacts. A close derivative of it was implemented by Japan Radio Co. for inclusion in a 3D ultrasound machine. More recently, VolumePro 1000, built on its predecessor VolumePro 500, was introduced. It can render up to $512 \times 512 \times 512$ volumes in real time with improved image quality. It also supports the mixing of polyfgons and volumes.

The choice of whether one adopts a general- or special-purpose solution to volume rendering depends upon the circumstances. If maximum flexibility is required, general-purpose appears to be the best way to proceed. However, an important feature of graphics accelerators is that they are integrated into a much larger environment where software can shape the form of input and output data, thereby providing the additional flexibility that is needed. A good example is the relationship between the needs of conventional computer graphics and special-purpose graphics hardware. Nobody would dispute the necessity for polygon graphics acceleration despite its obvious limitations. The exact same argument can be made for special-purpose volume rendering architectures.

6. Conclusions

The progress so far in volume visualization techniques, in hardware, and memory systems, coupled with the desire to reveal the inner structures of volumetric objects, suggests that volume visualization and volume graphics may develop into major trends in computer graphics. Just as raster graphics in the seventies superseded vector graphics for visualizing surfaces, volume visualization and volume graphics have the potential to supersede surface graphics for handling and visualizing volumes as well as surfaces.

Acknowledgments

This work was partially supported by ONR grant N000140110034 and NYSTAR C000057. Thanks are due to my PhD students and post-docs who contributed the work summarized in this paper.

7. References

1. Bakalash, R., Kaufman, A., Pacheco, R. and Pfister, H., "An Extended Volume Visualization System for Arbitrary Parallel Projection", *Proceedings of the 1992 Eurographics Workshop on Graphics Hardware*, Cambridge, UK, September 1992.

2. Bitter, I. and Kaufman, A., "A Ray-Slice-Sweep Volume Rendering Engine", *SIGGRAPH-Eurographics Workshop on Graphics Hardware Proceedings*, August 1997, 121-130, 155.

3. Cohen-Or, D., Rich, E., Lerner, U. and Shenkar, V., "A Real-Time Photo-Realistic Visual Flythrough", *IEEE Transactions on Visualization and Computer Graphics*, **2**, 3 (September 1996), 255-265.

4. Cohen-Or, D. and Kaufman, A., "3D Line Voxelization and Connectivity Control", *IEEE Computer Graphics & Applications*, **17**, 6 (November 1997), 80-87.

5. de Boer, M., Groepl, A., Hesser, J. and Maenner, R., "Latency- and Hazard-Free Volume Memory Architecture for Direct Volume Rendering", *Computers & Graphics*, **21**, 2 (1997), 179-187.

6. Goldwasser, S. M., Reynolds, R. A., Bapty, T., Baraff, D., Summers, J., Talton, D. A. and Walsh, E., "Physician's Workstation with Real-Time Performance", *IEEE Computer Graphics & Applications*, **5**, 12 (December 1985), 44-57.

7. Guenther, T., Poliwoda, C., Reinhard, C., Hesser, J., Maenner, R., Meinzer, H. and Baur, H., "VIRIM: A Mssivealy Parallel Processor for Real-Time Volume Visualization in Medicine", *Proceedings of the 9th Eurographics Hardware Workshop*, Oslo, Norway, September 1994, 103-108.

8. Jackel, D., "The Graphics PARCUM System: A 3D Memory Based Computer Architecture for Processing and Display of Solid Models", *Computer Graphics Forum*, **4**, (1985), 21-32.

9. Kaufman, A. and Shimony, E., "3D Scan-Conversion Algorithms for Voxel-Based Graphics", *Proc. ACM Workshop on Interactive 3D Graphics*, Chapel Hill, NC, October 1986, 45-76.

10. Kaufman, A., "Efficient Algorithms for 3D Scan-Conversion of Parametric Curves, Surfaces, and Volumes", *Computer Graphics*, **21**, 4 (July 1987), 171-179.

11. Kaufman, A., "An Algorithm for 3D Scan-Conversion of Polygons", *Proc. EUROGRAPHICS'87*, Amsterdam, Netherlands, August 1987, 197-208.

12. Kaufman, A. and Bakalash, R., "Memory and Processing Architecture for 3-D Voxel-Based Imagery", *IEEE Computer Graphics & Applications*, **8**, 6 (November 1988), 10-23. Also in Japanese, *Nikkei Computer Graphics*, 3, 30, March 1989, pp. 148-160.

13. Kaufman, A., *Volume Visualization*, IEEE Computer Society Press Tutorial, Los Alamitos, CA, 1991.

14. Kaufman, A., Cohen, D. and Yagel, R., "Volume Graphics", *IEEE Computer*, **26**, 7 (July 1993), 51-64. Also in Japanese, *Nikkei Computer Graphics*, 1, No. 88, 148-155 & 2, No. 89, 130-137, 1994.

15. Knittel, G. and Strasser, W., "A Compact Volume Rendering Accelerator", *Volume Visualization Symposium Proceedings*,

Washington, DC, October 1994, 67-74.

16. Levoy, M., "Display of Surfaces from Volume Data", *IEEE Computer Graphics & Applications*, **8**, 5 (May 1988), 29-37.

17. Lorensen, W. E. and Cline, H. E., "Marching Cubes: A High Resolution 3D Surface Construction Algorithm", *Computer Graphics*, **21**, 4 (July 1987), 163-170.

18. Malzbender, T., "Fourier Volume Rendering", *ACM Transactions of Graphics*, **12**, 3 (July 1993), 233-250.

19. Meagher, D. J., "Applying Solids Processing Methods to Medical Planning", *Proceedings NCGA'85*, Dallas, TX, April 1985, 101-109.

20. Meissner, M., Kanus, U. and Strasser, W., "VIZARD II, A PCI-Card for Real-Time Volume Rendering", *Proceedings of the SIGGRAPH/Eurographics Hardware Workshop*, August 1998, 61-67.

21. Muraki, S., "Volume Data and Wavelet Transform", *IEEE Computer Graphics & Applications*, **13**, 4 (July 1993), 50-56.

22. Ohashi, T., Uchiki, T. and Tokoro, M., "A Three-Dimensional Shaded Display Method for Voxel-Based Representation", *Proceedings EUROGRAPHICS '85*, Nice, France, September 1985, 221-232.

23. Osborne, R., Pfister, H., Lauer, H., McKenzie, N., Gibson, S., Hiatt, W. and Ohkami, H., "EM-Cube: an Architecture for Low-Cost Real-Time Volume Rendering", *Proceedings of the SIGGRAPH/Eurographics Hardware Workshop*, 1997, 131-138.

24. Pfister, H., Kaufman, A. and Chiueh, T., "Cube-3: A Real-Time Architecture for High-resolution Volume Visualization", *Volume Visualization Symposium Proceedings*, Washington, DC, October 1994, 75-82.

25. Pfister, H. and Kaufman, A., "Cube-4: A Scalable Architecture for Real-Time Volume Rendering", *Volume Visualization Symposium Proceedings*, San Francisco, CA, October 1996, 47-54.

26. Pfister, H., Hardenbergh, J., Knittel, J., Lauer, H. and Seiler, L., "The VolumePro Real-Time Ray-Casting System", *Proceedings SIGGRAPH'99*, August 1999, 251-260.

27. Sramek, M. and Kaufman, A., "Alias-free Voxelization of Geometric Objects", *IEEE Transactions on Visualization and Computer Graphics*, **5**, 3 (July-Sept. 1999), 251-267.

28. Wan, M., Qu, H. and Kaufman, A., "Virtual Flythrough over a Voxel-Based Terrain", *IEEE Virtual Reality Conference Proceedings*, March 1999, 53-60.

29. Wang, S. and Kaufman, A., "Volume Sampled Voxelization of Geometric Primitives", *Visualization '93 Proceedings*, San Jose, CA, October 1993, 78-84.

30. Westover, L., "Footprint Evaluation for Volume Rendering", *Computer Graphics*, **24**, 4 (August 1990), 367-376.

31. Wright, J. and Hsieh, J., "A Voxel-Based, Forward Projection Algorithm for Rendering Surface and Volumetric Data", *Proceedings Visualization '92*, Boston, MA, October 1992, 340-348.

32. Yeo, B. and Liu, B., "Volume Rendering of DCT-Based Compressed 3D Scalar Data", *IEEE Transactions on Visualization and Computer Graphics*, **1**, 1 (March 1995), 29-43.

0-7695-1674-2/02 $17.00 © 2002 **IEEE**